

<u>DB Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
USPT,JPAB,EPAB,DWPI,TDBD	l4 same (web near2 server)	4	L16
USPT,JPAB,EPAB,DWPI,TDBD	l14 same request	13	L15
USPT,JPAB,EPAB,DWPI,TDBD	l5 same server	37	L14
USPT,JPAB,EPAB,DWPI,TDBD	l2 same l4	1	L13
USPT,JPAB,EPAB,DWPI,TDBD	l4 and l5	2	L12
USPT,JPAB,EPAB,DWPI,TDBD	l8 and l4	2	L11
USPT,JPAB,EPAB,DWPI,TDBD	l8 same l4	1	L10
USPT,JPAB,EPAB,DWPI,TDBD	l8 same l3	18	L9
USPT,JPAB,EPAB,DWPI,TDBD	l1 same l2	57	L8
USPT,JPAB,EPAB,DWPI,TDBD	l2 same l5	2	L7
USPT,JPAB,EPAB,DWPI,TDBD	l2 and l5	12	L6
USPT,JPAB,EPAB,DWPI,TDBD	((dynamic\$ or automatic\$) near3 generat\$ near3 (web near2 page))	69	L5
USPT,JPAB,EPAB,DWPI,TDBD	((concurrent\$ or simultaneous\$) near2 process\$ near3 request)	378	L4
USPT,JPAB,EPAB,DWPI,TDBD	(process\$ near3 request)	25643	L3
USPT,JPAB,EPAB,DWPI,TDBD	(intercept\$ near3 request)	634	L2
USPT,JPAB,EPAB,DWPI,TDBD	(rout\$ near3 request)	5486	L1

WEST**End of Result Set**

Generate Collection

L13: Entry 1 of 1

File: EPAB

Oct 30, 1997

PUB-NO: WO009740617A1

DOCUMENT-IDENTIFIER: WO 9740617 A1

TITLE: TITLE DATA NOT AVAILABLE

PUBN-DATE: October 30, 1997

ABSTRACT:

The present invention teaches a system for creating and managing custom Web sites, specifically, managing a dynamic Web page generation request from a Web client (200) to a Web server (201). Instead of Web server executable (201(E)) processing the request, Interceptor (400) intercepts the request and routes it to Dispatcher (402). Dispatcher (402) receives the intercepted request, examines the request, and dispatches the request to one of a number of Page servers (404). The specified Page server (404) processes the request while Web server executable (201(E)) concurrently process other Web client requests.

WEST

Generate Collection

L15: Entry 1 of 13

File: USPT

Feb 6, 2001

DOCUMENT-IDENTIFIER: US 6185614 B1

TITLE: Method and system for collecting user profile information over the world-wide web in the presence of dynamic content using document comparators

ABPL:

Disclosed is a method and system for collecting profile information about users accessing dynamically generated content from one or more servers. In a specific embodiment, a server dynamically generates a web page in response to a user request. The server customizes the web page content based on the requested universal resource identifier (URI) and one or more of: the user's identity, access permissions, demographic information, and previous behavior at the site. The web server then passes the URI, user identity, and dynamically generated web page to an access information collector. The access information collector generates document comparators from the current web page content and compares them to document comparators associated with previously retrieved web pages. If the current web page is sufficiently similar to some previously retrieved web page, the access information collector logs the URI, user identity, and a document key associated with the matching previously retrieved page. Otherwise, the access information collector generates a new key; stores the new key and the document comparators in a database; and logs the URI, user identity, and the newly generated document key.

WEST☐ Generate Collection

L16: Entry 1 of 4

File: USPT

Oct 31, 2000

DOCUMENT-IDENTIFIER: US 6141759 A

TITLE: System and architecture for distributing, monitoring, and managing information requests on a computer network

DEPR:

A typical Request Broker configuration file is shown in FIGS. 5a through 5d. The configuration file contains a number of categories of information related to the Request Broker 90 operation. For example, category "Broker Configuration" 79 contains a number of variable values, such as AdapterEnable, MaxConnections, MaxContentServers, LocalFallback, and UseLocalPool. The "AdapterEnabled" value indicates whether the Request Broker 90 is enabled to handle requests from the web server program 64. The "MaxConnections" value indicates the maximum number of concurrent communications sessions (e.g., TCP/IP sockets) to be opened by the Request Broker 90 with any Application Server 92. The "MaxContentServers" value indicates the maximum number of local concurrent request processes that will be attempted on the Request Broker 90. The "LocalFallback" value indicates whether the pool of local data sources on the Request Broker 90 will be used when the remote network Application Servers 72,74 fail. The "UseLocalPool" value indicates whether the Request Broker 90 will act as a local application server by using the pool of local data sources in conjunction with the remote network Application Servers 72,74.

WEST

Generate Collection

L16: Entry 3 of 4

File: USPT

May 19, 1998

DOCUMENT-IDENTIFIER: US 5754772 A

TITLE: Transaction service independent HTTP server-to-transaction gateway

BSPR:

An on-line transaction processing system is made accessible to Web Browsers by establishing a predetermined plurality of transaction gateway clients to receive HTTP requests that are received by a Web Server from the Web Browsers. Concurrent processing of multiple transaction requests from the Web Browsers is performed by the plurality of transaction gateway clients. Each transaction gateway client pre-establishes a static connection with the on-line transaction processing system. The pre-established connection allows requests from the Web Browsers to be quickly routed to the transaction processing system. Time is saved by elimination of the traditional steps of connecting with and then disconnecting from the transaction processing system for each request from a browser program. The gateway client translates between HTTP formatted requests from the Web Browsers and the request format expected by the on-line transaction processing system.

BSPR:

The invention handles multiple concurrent requests from the Web Browsers and makes the requests available for concurrent processing by the on-line transaction processing system. A predetermined number of instances of the transaction gateway client are established to be available for performing the necessary translations. Each of the instances of the transaction gateway client establishes a static connection with the on-line transaction processing system as described above. As requests are received by the Web Server from the Web Browsers, the requests are routed to an available one of the instances of the transaction gateway client. Each instance of the transaction gateway client is capable of processing one request at a time.

WEST

Generate Collection

L6: Entry 3 of 12

File: USPT

Sep 12, 2000

DOCUMENT-IDENTIFIER: US 6118768 A

TITLE: Apparatus and methods for use therein for an ISDN LAN modem utilizing browser-based configuration with adaptation of network parameters

BSPR:

Specifically, in accordance with specific teachings of the present invention, once the workstation is connected to the hub and the browser begins executing on the workstation, the LAN modem automatically adapts itself to the current network environment of the workstation. To do so, the LAN modem will detect the Ethernet address of that workstation through packets transmitted by the workstation, determine the IP address of that workstation (either through dynamic assignment or by static address of the workstation from an ARP (address resolution protocol packet)), and then, if the workstation is using static addressing, set its own IP address and subnet mask such that the LAN modem and the workstation are on the same subnet. Once this occurs, the LAN modem and the workstation are then able to communicate over the network through the web browser. The LAN modem will then intercept any request issued by the workstation to fetch a web page and, through its own internal web server, will generate and download to the workstation, a default home page through which the user can commence configuring the LAN modem. Once configuration commences, the web server will then step the user through a succession of displayed web pages through which the user will be queried to enter salient configuration data. The web server will then extract this data from responses received from the user and then store this data, for subsequent use, in a shared database within the LAN modem.

DEPR:

Rather than maintaining a file store containing a file for each separate predefined web page in its entirety, particularly those containing graphics, which are then simply accessed--as is the case with conventional web servers and is costly in terms of memory, web server 412 constructs web pages in real-time. These pages are constructed from predefined stored templates (illustratively, for the preferred embodiment, approximately 600 bytes long) containing hypertext markup language (HTML) code that is common to all pages. For display of any one page, web server 412 simply accesses the stored code for the template and dynamically inserts appropriate predefined code segments therein in lieu of so-called "placeholder(s)" in the template based on a specific event that invoked display of that particular page. These segments can represent dialog boxes, graphics, predefined textual messages or, generically speaking, any object, whether HTML or otherwise, that needs to be selectively presented to a user either for display and/or to solicit a response, such as an item of data or a selection among a list of predefined data values, from the user. The manner through which code for such a template and all its associated objects is generated and the specific manner through which web pages are dynamically constructed therefrom are discussed in detail below in conjunction with File Creation process 2800 shown in FIG. 28 and illustrative code shown in FIGS. 30A-30B and 31. Since few full web pages are stored, memory requirements become rather modest. Collectively, the templates and all page components are stored within database 416 in flash memory 376 (see FIG. 3) and, in the preferred embodiment of the LAN modem, consume only approximately 200 Kbytes of storage space. Once a page is constructed by web server 412 (see FIG. 4B), a file for that page is then provided by the web server to HTTP process 415 which suitably packetizes and encapsulates that file, using the hypertext transfer protocol (HTTP). The resulting file is provided by HTTP process 415 to TCP/IP process 425 for eventual routing, over the LAN, to the associated workstation. User responses, in HTTP form, from the workstation to each web page are routed by TCP/IP process 425 to HTTP process 415 for suitable interpretation, such as

constructing and transmitting a next successive page to the user or storage of user-supplied configuration data. Web server 412 stores such user-supplied configuration data within database 416 for subsequent access and use. The architecture of web server 412 is discussed in considerable detail below in conjunction with FIGS. 18-21, with specific examples of dynamic web page generation being discussed below in conjunction with FIGS. 22-27.

DEPR:

Once the browser in the host, that issued the DNS query packet, receives this DNS reply packet, that browser will then issue a request to what it believes to be the remote DNS server to translate the domain name of a desired web server (typically that which stores a default web page defined in the browser) and then fetch a particular web page (e.g., the default page) therefrom. Though the browser will naturally assume that the IP address it received in the reply packet is that of a remote DNS server, in actuality it is that of the LAN modem itself--in effect the LAN modem has intercepted the DNS request from the host, i.e., a remote DNS server. TCP/IP process 425 (see FIG. 4B) within the LAN modem, in receiving this domain name translation request, will route this request to DNS server 421 which will return the IP address of the LAN modem itself. Once this occurs and the host issues a request to fetch the default page, web server 416, in response to receiving this request as routed to it by TCP/IP process 425, will test the shared (global) variable and determine, by the value of this variable, that an error condition has just occurred, and specifically the reason why an ISDN connection could not be completed then. Consequently, the web server, rather than returning a requested default page to the browser, will dynamically construct, through insertion of error-specific code segment(s) into a predefined page template, a web page that specifies an error condition has occurred, i.e., that an ISDN connection could not be established, and the specific reason why, e.g., the destination was busy or no B-channel was then available to accommodate the connection and then download this page, via TCP/IP process 425, to the host. The inventive manner through which this page is dynamically constructed to depict an error condition is discussed in detail below in conjunction with FIGS. 22-26.

DEPR:

Moreover, though configuration of the LAN modem has been described above in terms of communicating with an executing web browser in a workstation connected to the LAN modem, other appropriate TCP applications executing at the workstation and capable of conducting interactive communication with a server in the LAN modem and a user at the workstation, such as Telnet can be employed instead. Similarly, such applications can also be used in conjunction with interception of DNS Request (or other appropriate) messages to, e.g., a remote server, that would occur during TCP applications other than web browsing, to display appropriate fault messages.

DEPL:

c. Internal DNS And DHCP Servers And Interception Of Remote DNS Request For Error Handling

WEST☐ Generate Collection

L15: Entry 9 of 13

File: USPT

Jun 2, 1998

DOCUMENT-IDENTIFIER: US 5761673 A

TITLE: Method and apparatus for generating dynamic web pages by invoking a predefined procedural package stored in a database

ABPL:

A method and apparatus for generating dynamic Web pages is disclosed. Specifically, the present invention claims a method and apparatus for generating dynamic Web pages on a Web server by invoking and executing predefined procedural packages stored in a database. The claimed invention receives an object request on the Web server and activates a Web agent on the Web server based on the object request. The Web agent invokes and executes the predefined procedural package to retrieve data from a data repository, and then formats the retrieved data as HTML output.

BSPR:

Although Web pages were traditionally stored as static files on the Web server operating system, today Web pages can also be generated dynamically using the Common Gateway Interface (CGI). CGI is a standard interface for running external programs on a Web server. It allows Web servers to create dynamic documents when the server receives a request from the Web browser. When the Web server receives a request for a dynamic document, the Web server executes the appropriate CGI script and transmits the output of the execution back to the requesting Web browser. The Web browser does not differentiate between static and dynamic documents. It simply displays the output of the request.